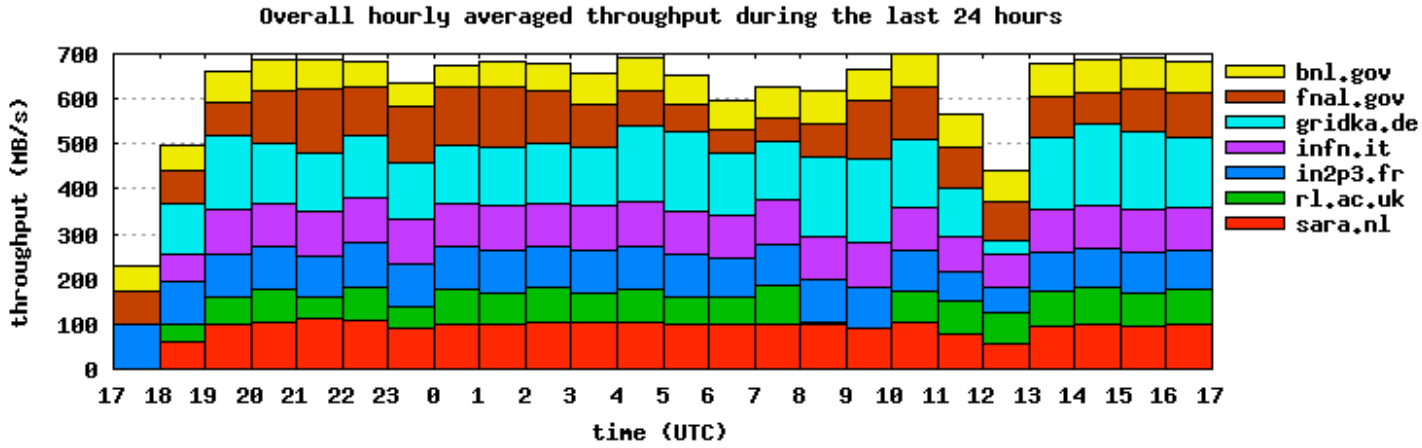
USCMS SC2 Summary Report Jon Bakken Preliminary Version 2, April 28, 2005.

These were the USCMS goals for SC2:

- Sustained PhEDEx driven and SRM-to-SRM managed transfers from Castor and openlab to FNAL dCache and onto tape via Enstore at a rate between 40 and 60 MB/s.
- Distributed SRM managed transfers from FNAL dCache to as many USCMS Tier2 sites as possible to demonstrate functionality.
- High rate, CERN disk to FNAL disk transfers, preferably managed by SRM to resilient dCache pools.

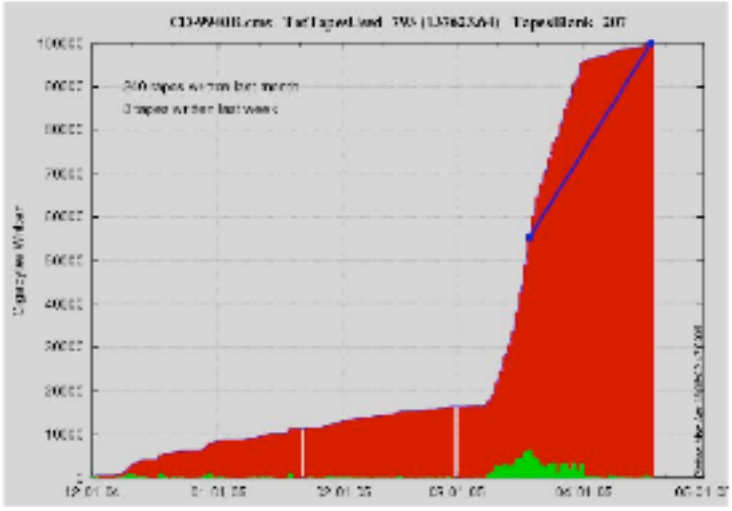
We were very successful on goals 1 and 3, and moderately successful for goal 2. Many people worked hard and helped us achieve these goals. Specifically, Timur P, Vladimir, P, and Andrey B provided almost all the changes documented below. Yujun W and Lassi T of the CMS dept worked extremely hard on many of the high level features of the challenge. Wayne B made sure tapes were recycled promptly. Finally, the whole cms-t1 team worked tirelessly during SC2 to keep things running properly.

FNAL throttled its rate to between 40 and 60 MB/s and wrote the data to tape. The data delivery rate to FNAL from CERN, just from the openlab cluster is shown in the following plot:



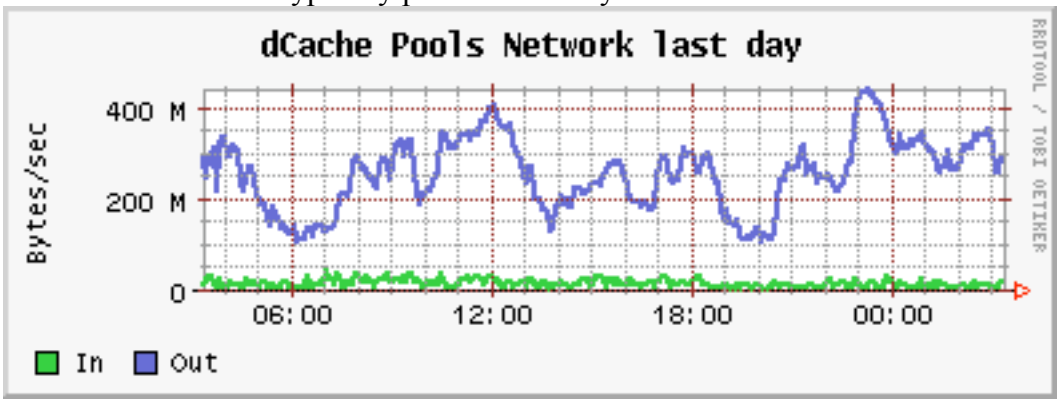
These graphs were available at http://challenge.matrix.sara.nl/SC, and were a great help in understanding overall SC3 performance from all sites. It was never a goal of FNAL to achieve a 700 MB/s aggregate bandwidth with all the sites combined.

All the data, both from the openlab and from castor were written to tape during SC3. A plot of tape usage is show in the following plot:

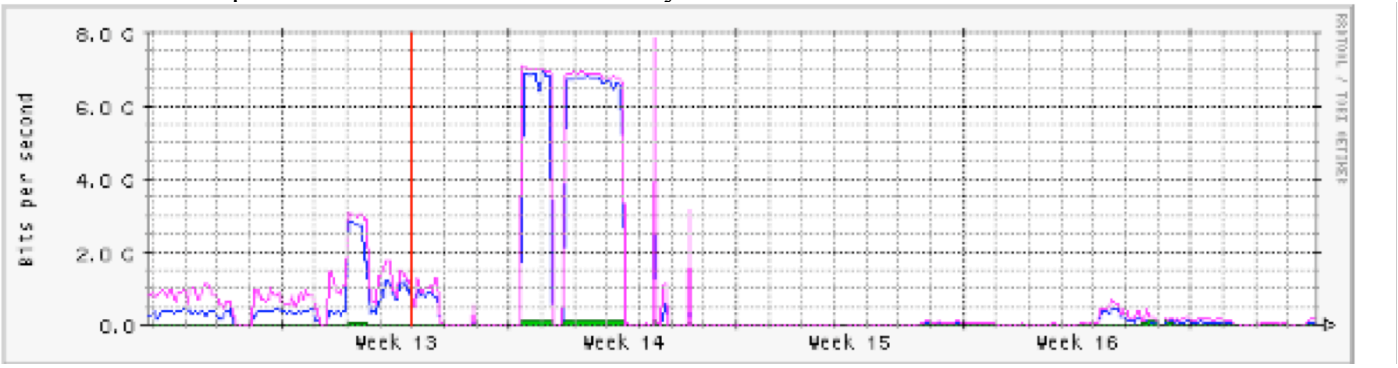


Once files were written, they were deleted from /pnfs, and the tapes were recycled twice a day. This was sustained for about 25 days.

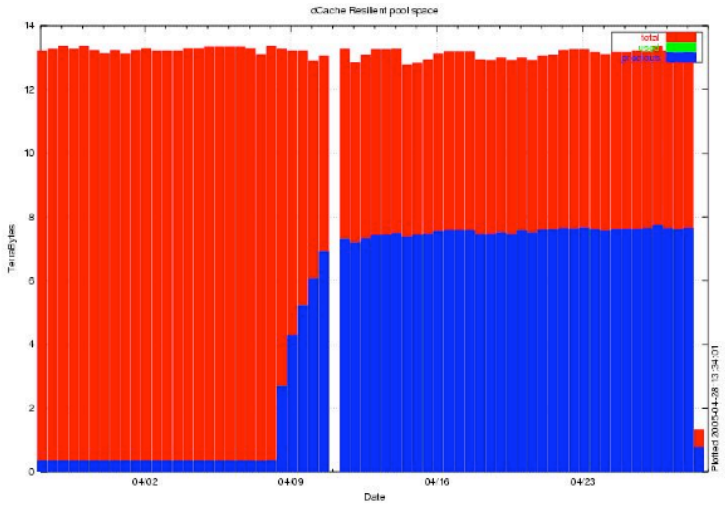
These extra 50 MB/s to tape were on top the normal user IO activity from/to the CMS production dCache pools. This typically varies between 200 and 400 MB/s. On one occasion during the challenge, the challenge and user IO was more than 1 GB/s. Here is a typically plot of user-only IO:



For the high-speed transfers, srmcp's from approximately 150 nodes requested data from CERN. This maxed out at ~700 MB/s. These files were written to resilient pool disks, but not managed via the SRM. None of this data was written to tape and files were deleted after they were written to the disk.

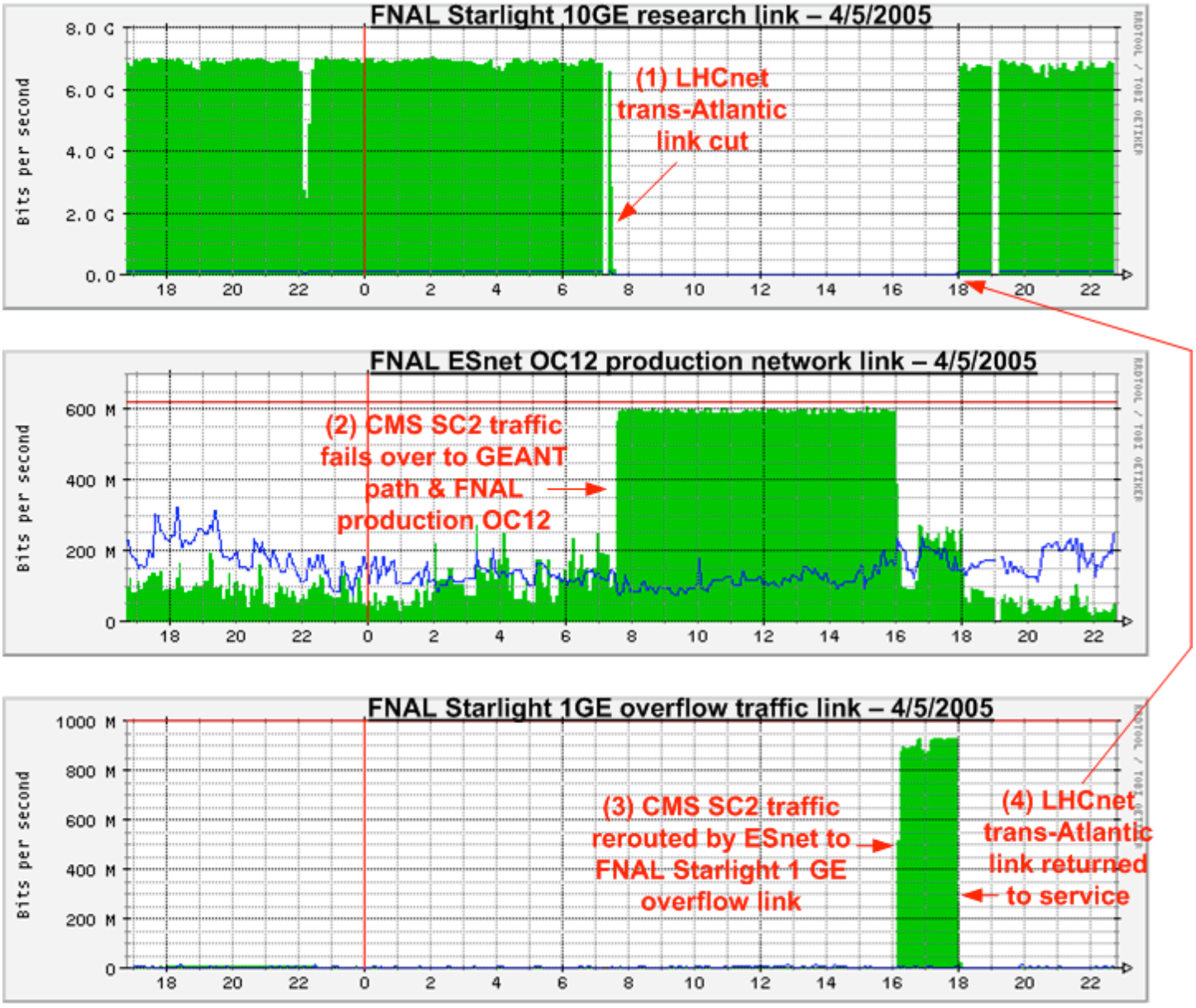


After running this for several days, CERN-SRM-to-FNAL-SRM managed transfer were started again as in the tape portion of the challenge, but this time the FNAL source were the resilient pools and the data was fully replicated by the resilient manager. This mode again maxed out at 700 MB/s. I only ran this last test for several hours since 700 MB/s fill up the resilient pools quickly. This barely shows up in the above plot – it is the narrow line at week 14, but it does show up well in the resilient pool space plot:



Deployment to Tier2 sites was accomplished, but it was a source of many problems and a large amount of frustration. The local dCache installed easily - it was the WAN deployment and resilient manager that were the problems. These two components are things developed at FNAL. Although the installation is now documented, it is not automatic and depends a great deal on the physical installation at remote sites. Much debugging was needed to get things working. University of Florida, who had their pool WAN interfaces on the public network (as we do at FNAL) worked early on and transferred files using PhEDEx. UCSD had their pools behind a NAT and the gridftp doors on the WAN network. Transfers worked early in the deployment, but stopped functioning later on - and the issue still has not been debugged properly. Caltech and University of Wisconsin's installation were never fully debugged, and didn't really participate in any PhEDEx transfers during SC2. Purdue installed everything correctly, essentially on the own, and participated in PhEDEx transfers at the tail end of SC2. After the challenge officially completed, Caltech started transferring data from FNAL via PhEDEx and managed SRM transfers at more than 10 MB/s.

There was one spectacular unplanned success: During the 700 MB/s portion of the tests, a trawler cut the transatlantic link. The CMS transfers were automatically rerouted to our ESNet link, albeit at 62.2 MB/s. The FNAL networking folks then manually rerouted it to the 1 GE link to alleviate the strain on ESNet and the transfer rate increased to the line's maximum of 100 MB/s. Later, service was restored to the 10 GE link and the transfers automatically rerouted to that default link and resumed at 700 MB/s. This is shown in this plot:



Accomplishments that helped us meet the SC2 goals:

- Before the challenge, paths from the CMS subnet to the openlab subnet at CERN were routed over our 10 GB starlight link. This involved negotiations between networking at CERN and FNAL. After SC2 it was found that CMS transfers to Castor were not on the starlight link - this was changed and all CMS transfers to openlab and Castor are now by default over starlight.

  The networking group also started to set up routes between FNAL and Tier2 sites. These efforts were mostly put on hold due to the lack of networking infrastructure at the Tier2 sites. This infrastructure is being planned or deployed now. Rate to Tier2 sites was not a goal of SC2, only functionality.

- All transfers were routed to pool gsiftp movers. These transfers went into the general dCache transfer queue. This was a significant problem. Normal user dccp traffic is moderate and typically only reads little information at a time. Compare this with gsiftp file transfers where 10-20 parallel streams are active, and you can see that the simple load throttling mechanisms available in the dCache can not handle this condition at all. There were 2 plans considered to overcome this.

  - Set up special pools for WAN traffic. This would work for writing into the cache, but was wasteful of resources. However, remote users also wanted to read data, and it was completely impractical for special resources to be set up for these transfers. Files are in one pool, and clients should be directed to where the data is, data should not be copied to special resources if at all possible.

  - Associate a cost with a transfer. Naively a dccp would have a cost of 1, and a gridftp transfer would be 10 or 20 (ore more) depending on the number of streams. This proved to be technically challenging, and ultimately deemed impracticable by DESY. DESY suggested, and is currently implementing, a hi-speed queue where all high speed transfers, like gridftp or file-based dccps, could be statically directed. This is a good compromise solution and should be ready for testing soon.

- It took 6 or more 9940B tape drives to sustain a 50 MB/s to tape rate. Each 9940B drive has a rate of 30 MB/s and streaming 1 GB files should give rates in the mid 20's, so 2-3 drives should have been sufficient for an input 50 MB/s rate. The problem was ultimately traced to simultaneous hi-speed read/write activity to the disks while encp was trying to write the data to tape. This resulted in poor Enstore encp disk read performance, and therefore poor 9940B drive rates.

  DESY observed this affect at their site as well. After discussion, DESY agreed to implement the solution both FNAL and DESY reached during investigations -- block incoming WAN writes from a group of pools and allow precious files to be drained via Enstore encp to tape while WAN transfers are suspended. After all files are on tape, another set of write pools would be chosen for draining and the current set would be re-opened for WAN writing. This solution is referred to as the smart HSM draining system and is currently being developed and should be available for testing shortly. Some local implementation details will still have to be worked out, but the method has worked in manual tests - now it 'just' has to be automated.

  Small files (castor data, not openlab) also dramatically affected both WAN rate, and Enstore tape writing rate. Small files should be avoided entirely.

- For SC2, CERN implemented DNS load balancing FTS (file transfer service) coupled with a backend DNS load balanced gridftp server. Our SRM clients had to be modified in order to use the round robin DNS FTS servers in 2 ways:

  - The FNAL SRM server had to be modified to handle multiple real SRM servers instead of just the typical single remote server.

- Once connected, the FNAL SRM server had to latch the actual name of the remote server so subsequent certificate exchanges between the server and client SRM server would authenticate properly.

  These changes allowed us to proceed with the SC2 transfers, but severely limit our ability to transfer data as we previously envisioned. Since all transfers go to single node once connected, only 1-2 files per session can be requested - otherwise all IO will be bound to a single remote node. The original plan had been to issue srmcp's with thousands of requests in them and let the SRM manage the transfers. The DNS based FTS system requires us to submit thousands of srmcp requests thereby increasing overhead.

  The entire DNS based gridftp design is flawed as well. It requires files to be copied locally to the node that is 'selected' by DNS rather than have clients directed to the nodes that already have the data. Besides this unnecessary copy, this design severely limits available local cache space. And, I do not know how it can work robustly without getting stalled in an environment of clients actively writing and reading data, especially when remote sites are down for extended periods. It seems the remote DNS-based FTS software is now a 'given', and this impacts us at FNAL. A careful re-design of how we interact must be undertaken.

- PhEDEx did manage to properly drive SRM-managed transfers. Once databases were loaded, a step that went quickly in some cases and slowly in others, the entire PhEDEx chain worked properly and reliably. I didn't understand why the loading of PhEDEx was problematic in some cases and easy in others. There are 2 issues that concern me:

  - Reliance of remote database in order to transfer data. We depend on the export site to determine what to transfer. It would seem more appropriate for this dependence to be local, with occasional remote updates, not the other way around.

  - Lack of automated graphs or charts indicating how the transfers are proceeding. Weekly graphs were made, but these were not helpful in debugging current situations

- An executive decision, put forth by Lothar B and accepted the CMS board all FNAL to transfer all data it holds onsite to its Tier2 sites, and if FNAL has space, it can request any data available at CERN without getting prior permission.

- About 3/4s of the way through SC2 we started severe slowdown in transfer requests. At one point, we were waiting for CERN to return a TURL to us. This was only observed for a few days, and never by the CERN folks when they investigated. The slowdown was then traced to the FNAL SRM server taking a long time to generate a transfer request. This was in turn traced to postgres. Several things were done to remove this effect, all based on advice from postgres experts.

  - Automated database cleaning was started. This was supposed to keep database lookups at constant time. [I don't understand this but it does appear to be real effect.] A cronjob was added to do this cleaning. This has been replaced by equivalent cleaning commands in the Java code itself.

  - The schema was examined again and all strings were converted to integers. This dramatically improved performance.

  - General code cleanup to optimize access.

  - Switch to postgres 8 for better performance.

- Independent postgres database for exclusive srm access. This step was mostly to eliminate other sources of slowdown. It can most likely be eliminated.

After these changes, the SRM again responded promptly and transfers were started soon after the PhEDEx script requested them.

- Significant effort was expended in getting the Tier2 sites running. The default dcache.org dCache was installed at each site. Then the resilient manager was installed on top of this. Further, the pnfs databases at the tier2s were converted to postgres to prevent size limitations and increase performance.

  Debugging remote sites was exceedingly difficult. We didn't know the configuration of the remote sites exactly, and each one was different. Errors fell into 2 general categories: (a) troubles with WAN connections due primarily to NATs and firewalls, and (b) remote site admin mistakes. [We all make mistakes, but things like putting incorrect IPs in /etc/hosts are hard to find, and ultimately have nothing to do with the dCache installation, other than assuring it won't work.]

- At the beginning of SC2, understanding traffic between FNAL and Tier2 sites was very difficult. I was doing netstats and tcpdumps and never really understood the high level interactions and rates. PhEDEx didn't provide any dynamic feedback to transfer rates at all other than a single transfer rate number. It would seem prudent to have this better integrated into Mona Lisa, or similar, monitoring frameworks.

- The networking group at FNAL provided easy to use detailed tools that completely solved this problem. Their tools are tabulated at http://www-dcn.fnal.gov/~netadmin/CMS/index.html and provided the following capabilities:

  - Multi-stream and aggregate rates for each gridftp transfer. This allowed us to maximize throughput based on the buffer size and the number of streams.

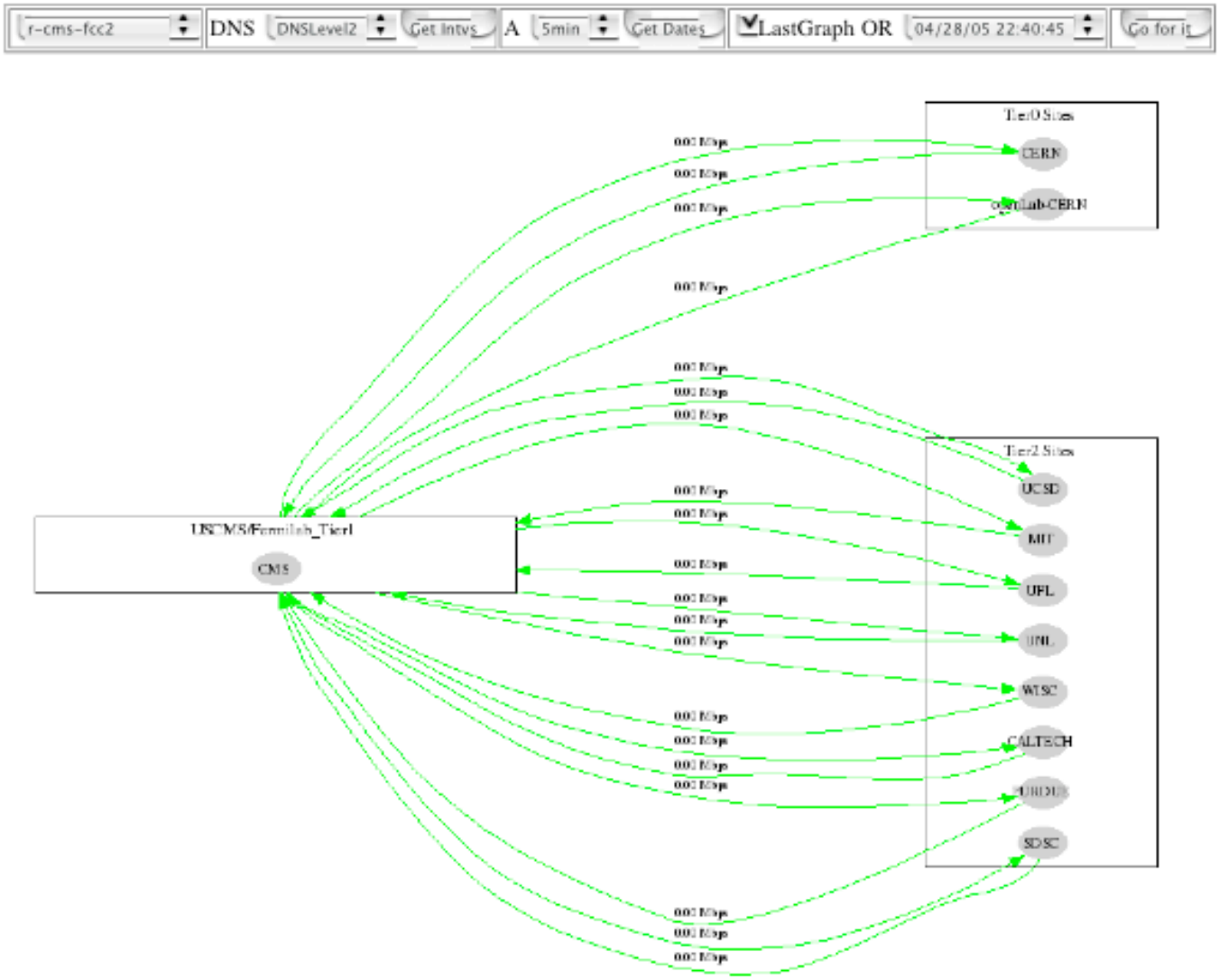
  - Real-time global picture of transfers CERN<->FNAL<->Tier2 as shown in plot on the next page. This picture significantly added to our understanding of where transfers were going and at what rates. Without it, I was evaluating things locally -- this allowed a great global view of transfers to/from the FNAL site.

  - Detailed real-time mrtg-like plots for CERN<->FNAL<-Tier2 transfers These plots were like the CERN-FNAL Starlight MRTG graphs, but for FNAL<->Tier2 sites. They allowed me to view the historical rates to each site.

  - Representation of traffic for CMS end-to-end transferring in the form of tables with breakdown of involved hosts based DNS level 0,1,2.. These tables allowed me to immediately see who was accessing the CMS dCache. This is how I found out that Purdue was ready and even transferring data, for example.

I want to emphasize the importance of this networking information, all available via web interfaces.. It provided great understanding of the global picture, something that was missing at the beginning of SC2. I do not know how this kind of work gets transferred to other sites, or if other sites will have their own infrastructure in place.

The map of traffic for Tier2 at 04/28/05 22:40:45, Time now is Thu Apr 28 22:48:12 2005 SAB.



- The gridftp door became unreliable at FNAL and at UCSD. It had to be restarted several times at FNAL. This is mostly due to too many active transfers, but remains an open issue and is being worked on.

- Transfers RAL<--FNAL, initiated by RAL failed. We investigated and found inconsistencies in our logs. We asked the user for corresponding entries from the RAL logs. The user responded he didn't have access to them. This ended the investigations. Tier1 to Tier1 transfers need to be worked out by facility groups not by end users.

- Error messages in the code were changed from, for example, "do not" to "don't". This seemingly harmless effect caused the postgres evaluator to end its interpretation on the apostrophe in don't, causing many database lookups to fail. The SRM was changed to quote these embedded apostrophes in all cases.

Finally, it should be noted that the user-driven rate was more than 45 MB/s for several days (castor to Enstore tape) and peaked at 70 MB/s for some periods. A lot of this data was shipped to several Tier2 sites.

Other significant accomplishments during SC2 or just after SC2, but not directly related to SC2 goals:

- The pnfs gdbm->postgres conversion was documented. This should help our tier2 sites with their own conversions.

- The FNAL SRM client and server were changed so they would be compatible with the LBL SRM/DRM server and clients. This was a problem in the wsdl expectations on both sites (I don't know more than that.) Compatibility with the CERN srmcp clients was also re-demonstrated.

- FAST was installed on 4 nodes in the CMS dCache pools. Although it wasn't specifically tested for improved TCP performance, the good news is that it is compatible with the current TCP implementation. The bad news is that it is incompatible with Enstore tape operations and had to be withdrawn. This remains an open issue.

- Space reservation for a single transfer was available. It was tried, but failed to work. Work is in progress to debug this badly needed option.

- x509 user proxy and number of parallel stream options in srmcp were added. This allowed increased user flexibility in controlling transfers.

- Enstore file database lookup to determine if files are on tape, rather than relying on pnfs for this information. This reduced pnfs load significantly, especially from PhEDEx scripts.

- An srmcp –report option was added to facilitate the parsing of errors from srmcp and to eliminate unneeded pnfs access by PhEDEx scripts. This also significantly reduced the load on pnfs. This option is akin to the --data-access-layer option in encp for the filling of the SAM databases in DO.

- 6509 IOS load balancing was shown to work. It required work by Andrey B, Dave F. and Matt C. Ultimately, it depends on a proper ssh client to be installed by users. The native mode OS has been loaded on the CMS 6509 switch and IOS load balancing is being phased in.

- Worker nodes (with resilient pools) were moved from FCC to GCC, and there were ongoing internal node shuffles between FCC1 and FCC2 during SC2.

- After SC2, a dataset with many small files in it was transferred from CERN to FNAL. All transfers at the USCMS Tier1 facility started to fail due to an extreme pnfs slowdown. This was traced to SRM server or client problems in transferring files that reside in directories with a large number of files in the (more than 45000). This in turn was traced to a pnfs manager reverse lookup problem on file creation. DESY is working on a solution to this pressing problem. However, it’s clear that directories should be limited to no more than 5000 files. Users want to ls directories occasionally, and large number of files are always problematic.

-